IJoFCS (2011) 1, 28-43

e-forensic PRESS

The International Journal of FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

DOI: 10.5769/J201101002 or http://dx.doi.org/10.5769/J201101002

Acquisition and Analysis of Digital Evidence in Android Smartphones

André Morum de L. Simão[1], Fábio Caús Sícoli[1], Laerte Peotta de Melo[2], Flávio Elias de Deus[2], Rafael Timóteo de Sousa Júnior[2]

(1) Brazilian Federal Police, Ministry of Justice
(2) University of Brasilia, UnB
(1,2) Brasilia, Brazil
(1) {morum.amls, sicoli.fcs}@dpf.gov.br
(2) {peotta, flavioelias, desousa}@unb.br

Abstract - From an expert's standpoint, an Android phone is a large data repository that can be stored either locally or remotely. Besides, its platform allows analysts to acquire device data and evidence, collecting information about its owner and facts under investigation. This way, by means of exploring and cross referencing that rich data source, one can get information related to unlawful acts and its perpetrator. There are widespread and well documented approaches to forensic examining mobile devices and computers. Nevertheless, they are neither specific nor detailed enough to be conducted on Android cell phones. These approaches are not totally adequate to examine modern smartphones, since these devices have internal memories whose removal or mirroring procedures are considered invasive and complex, due to difficulties in having direct hardware access. The exam and analysis are not supported by forensic tools when having to deal with specific file systems, such as YAFFS2 (Yet Another Flash File System). Furthermore, specific features of each smartphone platform have to be considered prior to acquiring and analyzing its data. In order to deal with those challenges, this paper proposes a method to perform data acquisition and analysis of Android smartphones, regardless of version and manufacturer. The proposed approach takes into account existing techniques of computer and cell phone forensic examination, adapting them to specific Android characteristics, its data storage structure, popular applications and the conditions under which the device was sent to the forensic examiner. The method was defined in a broad manner, not naming specific tools or techniques. Then, it was deployed into the examination of six Android smartphones, which addressed different scenarios that an analyst might face, and was validated to perform an entire evidence acquisition and analysis.

Keywords - forensic analysis, data acquisition, evidence analysis, cell phone, smartphone, Android.

I. Introduction

In 2011, the Android operating system has exceeded the number of handsets sold in other systems for smartphones [1]. According to Gartner [2], the system has a wide acceptance in the market, as it hit 52.5% of the worldwide market share in the third quarter of the year. The platform's success may be due to being open source and supporting the latest features and applications available for this type of mobile equipment. Given its ability to provide a large number of features to the user, a smartphone with the Android operating system can store a significant amount of information about its owner, being a source of evidence for facts one wants to clarify or to obtain information to support an investigation [3].

Unlike the data acquisition approach for computer environments, when data can usually be extracted in the same state they were found and is preserved from the time of their seizure, data extraction from smartphones typically requires intervening on the device. Moreover, given that they use embedded memories, whose direct hardware access is delicate and complex, sometimes there is a need to install applications or use tools directly on the device to proceed with the stored data acquisition. Thus, the analyst must have the expert knowledge required to carry out forensic procedures on the device the least intrusive manner as possible, controlling the environment in order to avoid loss, alteration or contamination of evidence data [4], which will give reliability to the forensic procedure.

II. Brief History of Mobile Phones

The Figure 2 shows the evolution of cell phones. Over the years, it demonstrates the growing need for users to add more and more information on those devices. It leads to the moment experienced today with the advent of smartphones, when Android devices play an important role in the evolution of this technology.

The first cell phone prototype was developed by Martin Cooper [5] and introduced by Motorola in 1973. It was called DynaTAC 8000X, had approximately 30 cm and weighed almost a pound. It was only available for sale ten years later, in 1983, for US$ 3,995.00. At that time, it had a battery capable of one hour of talk time and memory to store 30 telephone numbers [6].

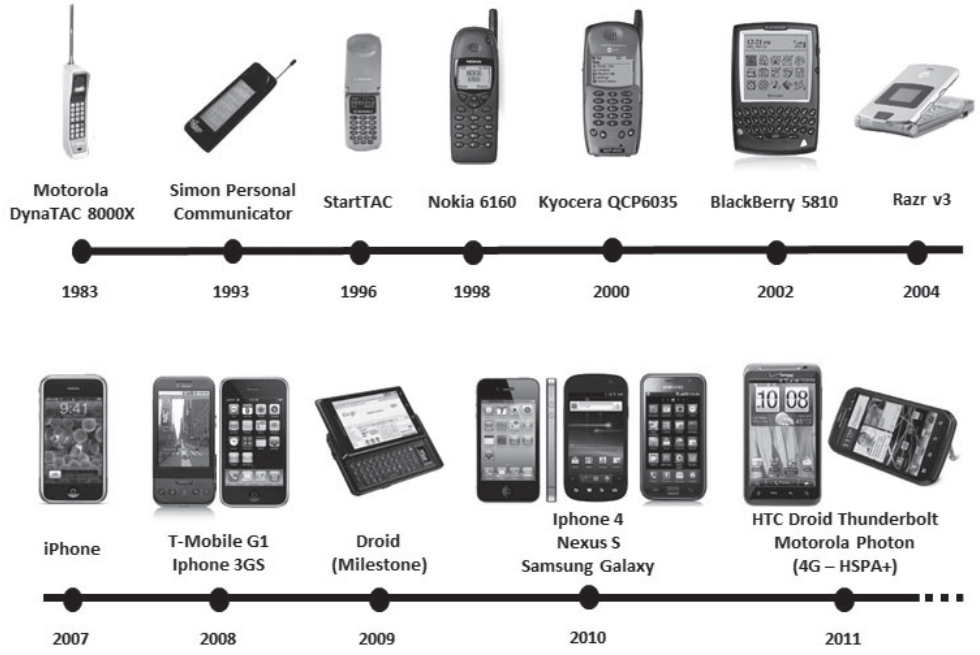

Figure 1. Cell Phones evolution

In 1993, IBM (Bell South) introduced the first mobile phone with PDA features (Personal Data Assistant), the “Simon Personal”. In 1996, Motorola presented its model "StarTac", with only 87 grams, which was a success because it had many desirable features (calendar, contacts and caller ID) and was a device with a distinctive aesthetic, which led to a meaningful market success. Nokia was successful with its phones which had an innovative design (candybar-style) at the end of the 1990’s with the launch of the Nokia 6160 in 1998, weighing 170 grams, and the 8260 in 2000, with 96 grams [6]

The first mobile phone to have the Palm operating system was introduced by Kyocera in 2000, the model "QCP6035".

In 2002, it was presented by the American company Danger Hiptop (T-Mobile Sidekick), one of the first cell phones with a web browser, e-mail and instant messaging [7]. In the same year, the Research In Motion (RIM) company launched the "BlackBerry 5810" with features of electronic messaging, personal organizer, calendar and physical keyboard.

The following year, Nokia launched the "N-Gage," which was a mobile phone that also functioned as a handheld game console. Motorola invested huge amounts in its cell phone design and had even more success in its slim mobile phone "RAZR V3" in 2004. The phone was considered very popular, since it was desired by people who had different uses for the device [6].

In 2007, the Apple company caused a great revolution in mobile phones by presenting the "iPhone” model. Those devices had a great computing power, portability and design, featuring today's standards, the so-called smartphones.

In 2008, the Open Handset Alliance (OHA) launched the mobile operating system Android. It was a response of the leaders in mobile phone market, such as Google, to Apple’s "iPhone". It featured a platform as functional as the competitor’s, but, as it was based on an open system, it was a cheaper alternative. The first phone with the Android operating system to be marketed was the "T-Mobile HTC G1" in 2008.

Since then, the industry-leading companies have made large investments in the smartphone platforms, boosting the market for mobile devices with more features that attract new users and retain those already familiar with the technology. With the release of the first Android smartphone, the market has seen a healthy battle between Apple devices (iPhone 3, 3GS, 4, 4S) and those with operating systems from Google (Nexus, Samsung Galaxy, Optimus LG, HTC Thunderbolt, Motorola Atrix/Phonton , and so on). In recent years, there has also been a great evolution in cellular networks, like the current high-speed 4G networks, providing more features to the users.

Without having success spreading their platforms, companies like Microsoft and Nokia united their efforts in 2011 to try to gain more space in the smartphone market dominated by Apple and Android [8]. Their Windows Phone (formerly Windows CE) and Symbian operating systems were being considered outdated.

III. Android Platform

Android is an open operating system designed for use on mobile devices. The world-renowned company, Google Inc. bought Android Inc. in 2005, hiring Andy Rubin as director of mobile platforms group [9]. On November 5th, 2007, the Open Handset Alliance (OHA), which is a consortium of over 80 major companies in the mobile market, such as Motorola, Samsung, Sony Ericsson and LG, was founded and has invested and contributed to the Android platform development. The source code for Android is released under the Apache License, Version 2.0.

The Android platform is basically composed by the operating system, the SDK (Software Development Kit) and applications. The SDK is a set of tools provided by Google that provides a development environment for creating Android compatible software. Android applications use the Java programming language, which is wide-

spread and accepted. For reasons that go beyond this paper, Google chose not to use the standard Java platform, and picked the Dalvik virtual machine (DVM - Dalvik Virtual Machine) instead.

Currently, the Android operating system is commercially available in 7 (seven) versions: 1.5 (cupcake), 1.6 (donut), 2.0/2.1 (eclair), 2.2 (Froyo), 2.3 (gingerbread), 3.0/3.1/3.2 (honeycomb - dedicated exclusively to the tablet PC market) and 4.0 (Ice Cream Sandwich). The work presented in this paper shall not apply to the last two versions, since the 3.x is not intended for smartphones and the 4.0 was recently released.

The software stack is divided into four layers, including five different groups, as shown in Figure 2.

The application layer consists of a basic set of applications, such as the web browser, electronic mail client, SMS program, calendar, contacts, map service, among others [10].

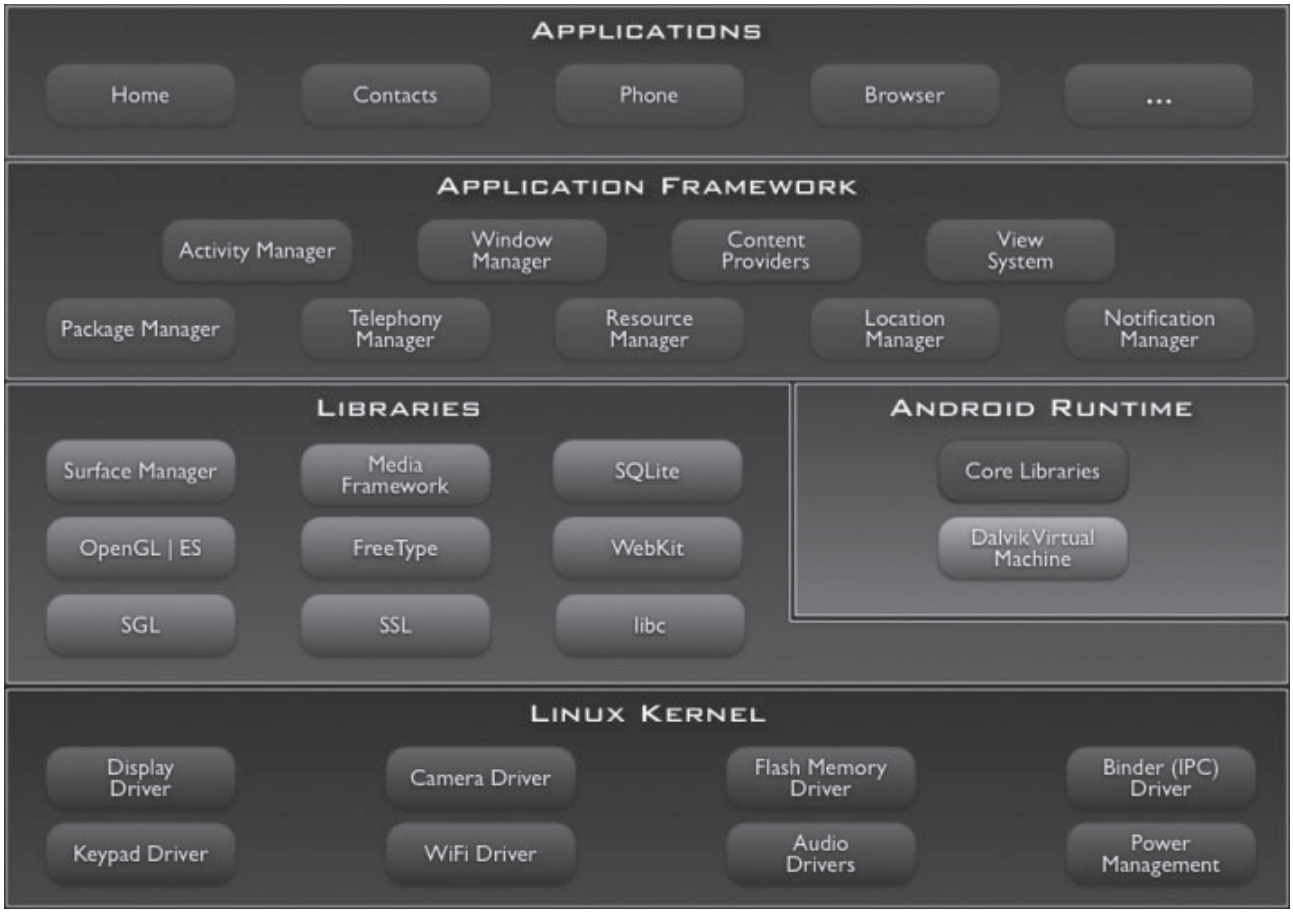

Figure 2. The components of the Android operating system [10]

The application framework provides an open and standardized development environment that allows, with the help of content providers and other services, the reuse of application functions and features. The whole API (Application Program Interface) available to the primary system is also available for application development, which provides developers with all available resources of the environment [10].

The libraries are written in C / C + + and invoked through a Java interface. The features offered by libraries are accessed through the application framework. Among the libraries are the ones to manage windows (surface manager), 2D and 3D media (codecs), and SQLite database to the web browser WebKit (used in Google Chrome and Apple Safari) [11].

The Android runtime environment has a set of libraries that provide all the features available in Java libraries on the operating system. These libraries enhance their features as Android versions are released. The Dalvik virtual machine works by interpreting and translating Java code into a language understood by the OS. It was developed in order to run multiple VMs efficiently, so that running binaries in Dalvik Executable format (.dex) could optimize memory usage [12].

The Linux 2.6 kernel is used by the Android operating system. It acts as an abstraction layer between the hardware and software stack and is responsible for device process management (driver model), memory management, network management and system security [13].

Regarding the file system, currently most of Android devices adopt YAFFS2 (Yet Another Flash File System 2), which is a file system designed for flash memory and its peculiarities. It is worth noting that the major forensic tools available are not compatible with that file system, making it difficult to mount Android partitions and access data stored there. However, as quoted by Andrew Hoog [14], in late 2010 it was observed that some Android handsets were already using the EXT4 (Fourth Extended File System). There is a migration tendency to this file system in order to support dual-core processor and multiprocessing, and to use e-MMC memories, (Embedded MultiMediaCard), which already work simulating block storage devices, that are more robust, mature and have more commercial acceptance.

The Android operating system uses the sandbox concept, where applications have reserved areas, with isolated process execution environments and limited access to resources. This way, applications cannot access areas that are not explicitly allowed [15]. However, access to features may be authorized by the permissions set in the "AndroidManifest.xml" file. At the time of application installation, that file tells the user what resources will be used on the smartphone. He can accept the installation of the application after being aware of the resources or simply refuse the installation, if he does not agree with the features that the application wishes to access.

Another feature of the Android OS is the use of the SQLite database, which is free and open source. It is an easy to use relational database, that stores in a single file the complete data object structure (tables, views, indexes, triggers) [16]. Such database does not need any configurations and uses file system permissions to control access to its stored data.

One of the tools available in the Android SDK is the Android Debug Bridge (ADB). It provides a communication interface to an Android system using a computer. When connected through this interface, a computer is able to access a command shell, install or remove applications, read log files, transfer files between the station and the device, among other actions.

Access to system partitions is restricted to the Android operating system. By default, users do not have permission to access system reserved areas. The system is shielded in order to prevent malicious or poorly developed applications to affect the OS's stability and reliability. However, it is possible to exploit a set of system or device vulnerabilities to obtain super user (root) privileges. Thus, it is possible to use applications or a shell that has full and unrestricted access to the system. As a result, a forensic analyst can make a mirror copy of all of the system partitions as well as access files which were not accessible by using Android conventional credentials. The techniques vary depending on each Android version and may also depend on the device manufacturer and model. Moreover, those techniques are often invasive and may even damage data stored on the device, so they should be used wisely.

The operating system has authentication mechanisms that use passwords, tactile patterns or biometric information. According to the NIST guide on cell phones forensics [17], there are three possible methods to unlock a device: investigative, software-based or hardware-based.

Those can be applied to Android equipment depending on the seizure circumstances, device model and system version.

Given the characteristics described, in order to conduct a forensic data extraction, besides having knowledge about the Android platform, an analyst should evaluate the procedures to be adopted. For instance, there are scenarios in which the phone may be turned on or off, have internal or removable memory, be locked or unlocked, have access through USB debug mode or not, have some applications running that contain useful information for an investigation, and even may have root privileges enabled. Thus, the analyst must assess the correct procedures to be adopted depending on the Android smartphone status.

IV. Data Acquisition Method for Android Smartphones

Considering the Android platform's unique characteristics and different scenarios which a forensic analyst may come across, a data acquisition method is proposed and its workflow is shown in Figure 3. In the figure, different scenarios are presented, along with the respective procedures that an analyst should perform. By using the proposed method, a forensic analyst may retrieve maximum information from the mobile device, so that the evidence may be documented, preserved and processed in the safest and least intrusive manner as possible.

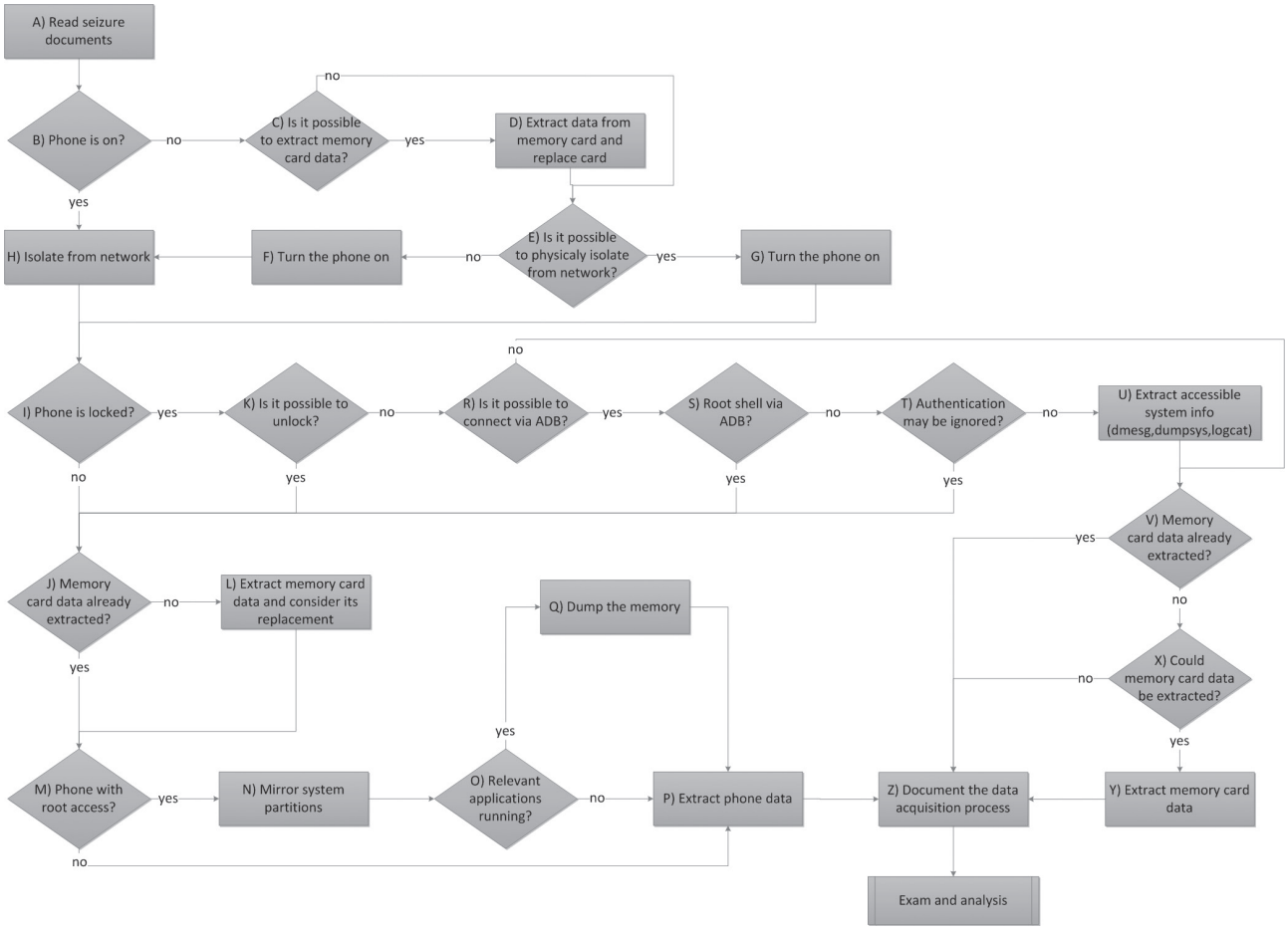

Figure 3. Workflow with the process of acquiring data from a smartphone with the Android operating system.

A. Initial procedures for data preservation in a smartphone

The Figure 4 illustrates the initial steps in data acquisition and preservation in Android devices. Upon receiving a smartphone, the forensic analyst must follow the procedures in order to preserve the data stored in the seized equipment. So, he should check if the phone is turned on or not. If the phone is powered off, one should evaluate the possibility of extracting data from its memory card. It should be pointed that some Android phones have an internal memory card, so it is not possible to remove it in order to copy its data through the use of a standard USB card reader. On the other hand, if it is feasible to detach the memory card, it should be removed and duplicated to an analyst memory card to ensure its preservation. To copy data from the memory card, one may use the same approach used with thumb drives. The forensic expert could use forensic tools to copy the data or even run a disk dump and then generate the hash of the duplicate data. At the end of the process, the analyst's memory card holding the copy should be returned to the device.

The next step is to isolate the telephone from telecommunication networks. The ideal situation is to use a room with physical isolation from electromagnetic signals. However, when one does not have such an infrastructure, he should set the smartphone to flight or offline mode. From the moment the power is on, he must immediately configure it to such connectionless mode, thus avoiding data transmission, receiving calls or SMS (Short Message Service) after the equipment seizure time. If by any chance, before it is isolated from the network, the phone receives an incoming call, message, email or other information, the analyst should document and describe it in his final report, which will be written after the data extraction, examination and analysis processes.

With the smartphone isolated from telecommunication networks, the forensic analyst should check if the Android has been configured to provide an authentication mechanism, such as a password or tactile pattern. Afterwards, he should carry out the procedures described in the following sections, which depend on the access control mechanism which is configured on the device.

B. Smartphone without access control

The least complex situation that an examiner may encounter is the one which the mobile is not locked and is readily able to have its data extracted. In this situation, one must first extract data from memory cards, if they have not been copied, and in case of removable memory cards, reinstall into the device the cards that have received the copies, preserving the original ones. Data acquisition processes of Android devices without access control are illustrated in Figure 5.

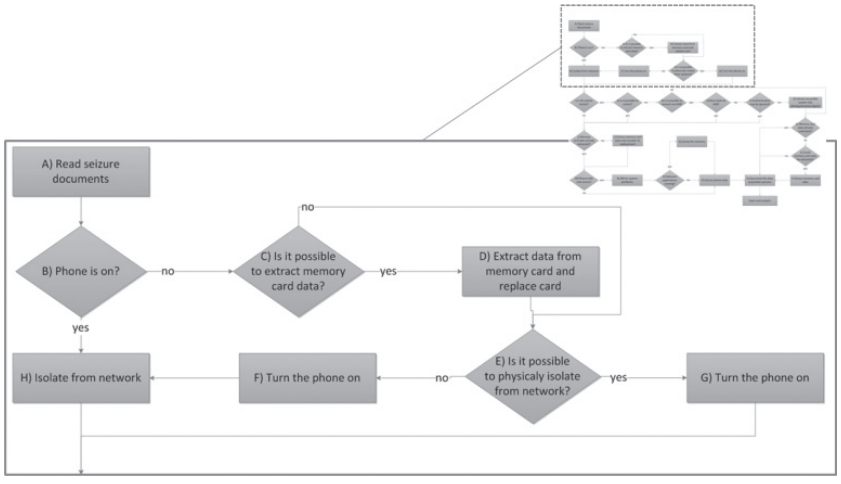

Figure 4. Initial procedures in data acquisition and preservation in Android devices.

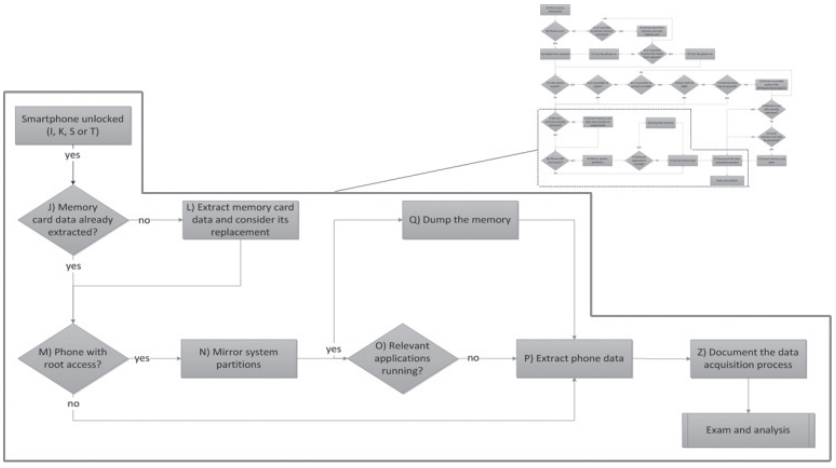

Figure 5. teps of data acquisition of an Android smartphone without access control.

With the data from memory cards extracted and properly preserved, the examiner should check if the Android has super user privileges enabled. The application called "Superuser" can be installed to provide access to such privileges. From the moment the analyst is faced with an Android phone with super user privileges, he can gain access to all data stored in the device without any restrictions. By using the USB debugging tool, ADB, present in the Android SDK, one can connect to the device, access a command shell with super user privileges and make a copy of the system partitions stored in its internal memory, as illustrated in Figure 6.

```
C:\Android\android-sdk\platform-tools>adb devices
List of devices attached
040140611301E014        device
C:\Android\android-sdk\platform-tools>adb -s 040140611301E014
shell
$ su -
su -
# mount | grep mtd
mount | grep mtd
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0
# cat /proc/mtd
cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
# ls /dev/mtd/mtd*
ls /dev/mtd/mtd*
…
/dev/mtd/mtd6
/dev/mtd/mtd6ro
/dev/mtd/mtd7
/dev/mtd/mtd7ro
/dev/mtd/mtd8
/dev/mtd/mtd8ro
…
# dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/mtd6ro_system.dd bs=4096
dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/mtd6ro_system.dd bs=4096
44960+0 records in
44960+0 records out
184156160 bytes transferred in 73.803 secs (2495239 bytes/sec)
# dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/mtd7ro_cache.dd bs=4096
dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/mtd7ro_cache.dd bs=4096
27136+0 records in
27136+0 records out
111149056 bytes transferred in 41.924 secs (2651203 bytes/sec)
# dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/mtd8ro_userdata.dd
bs=4096
dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/mtd8ro_userdata.dd bs=4096
50464+0 records in
50464+0 records out
206700544 bytes transferred in 74.452 secs (2776292 bytes/sec)
# ls /mnt/sdcard/*.dd
ls /mnt/sdcard/*.dd
mtd6ro_system.dd
mtd7ro_cache.dd

mtd8ro_userdata.dd
```

Figure 6. Commands to list connected devices, display partition information, and generate the partitions dump.

It should be pointed that, by carrying out the procedure described in Figure 6, the mirrored partition images will be written to the memory card which is installed in the device. In some situations, it may not be possible to replace the original memory card by an analyst's one. Nevertheless, regardless of its replacement, removable media's data must have been mirrored prior to system mirroring and copying. By doing that, data stored in the original memory card, seized with the smartphone, are preserved and the forensic expert should point that in his report that will be produced by the end of data analysis.

After mirroring the partitions, one should observe the running processes and assess the need to get run-time information, which is loaded in the

device's memory. Hence, it is possible to extract memory data used by running applications to access sensitive information, such as passwords and cryptographic keys. By using a command shell with super user credentials, the "/data/misc" directory's permissions must be changed. Afterwards, one must kill the target running process, so that a memory dump file for the killed process is created [18]. Data extraction of a telephone with available "super user" credentials may be finished in this moment. Figure 7 displays the technique described by Thomas Cannon [18].

```
# chmod 777 /data/misc
chmod 777 /data/misc
# kill -10 6440
kill -10 6440
# kill -10 6379
kill -10 6379
# kill -10 6199
kill -10 6199
# kill -10 5797
kill -10 5797
# ls /data/misc | grep dump
ls /data/misc | grep dump
heap-dump-tm1303909649-pid5797.hprof
heap-dump-tm1303909632-pid6199.hprof
heap-dump-tm1303909626-pid6379.hprof
heap-dump-tm1303909585-pid6440.hprof
#
…
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /
data/misc/heap-dump-tm1303909649-pid5797.hprof
2206 KB/s (2773648 bytes in 1.227s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /
data/misc/heap-dump-tm1303909632-pid6199.hprof
2236 KB/s (3548142 bytes in 1.549s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /
data/misc/heap-dump-tm1303909626-pid6379.hprof
1973 KB/s (3596506 bytes in 1.779s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /
data/misc/heap-dump-tm1303909585-pid6440.hprof
1968 KB/s (2892848 bytes in 1.435s)
```

Figure 7. Presents the commands to alter a directory's permissions, kill processes to create processes' memory dump files and copy those files to the analyst's station.

It is noteworthy that, in order to inspect the acquired data, the analyst should have an examination environment with tools that are capable of mounting images having the device's file system, which is in most cases the YAFFS2. The technique described by Andrew Hoog may be used to examine that file system [19]. Nevertheless, it is recommended that a logical copy of system files is made directly to the analyst's workstation, as shown in Figure 8.

```
C:\android-sdk\platform-tools> adb pull /data pericia/
Pull: building file list…
…
684 files pulled. 0 files skipped
857 KB/s (194876514 bytes in 226.941s)
```

Figure 8. Copy of logical files stored in the device's "/data" directory to the "pericia" directory in the analyst's workstation.

The data stored in the "/data" directory, for instance, contain information regarding the installed applications and system configuration. Logical copy of files will create redundancy that may be useful during the examination phase, especially in situations when it is not necessary to delve into system partitions. In addition, some applications may be active in the system, in such a way that a simple visual inspection may provide information which would be difficult to access by means of analyzing the created image. Moreover, forensic extraction tools may be used to interpret stored data.

In situations when the smartphone "super user" privileges are not available, data extraction from its internal memory should be carried out by visually inspecting and navigating the device's graphic user interface. Alternatively, forensic tools and applications may be used to assist the analyst in extracting device's data. Nevertheless, it is important to check the information gathered by such tools, because the Android OS has different versions, as well as manufacturer and telephone carrier customizations, which may interfere in the automated tools' proper functioning. There are numerous applications that may store meaningful information for an investigation, whose data extraction is not supported by forensic tools. It is clear that the forensic analyst needs to have proper knowledge regarding the Android platform and its applications, since relevant information extraction should be conducted in the most complete way.

Some Android smartphones allow their internal memories to be copied using boot loader or recovery partitions vulnerabilities, without having "super user" credentials. It is up to the analyst to evaluate if it is possible and viable to apply such techniques for that kind of device. It is suggested that the investigation team should discuss the need of using such procedures and consider the risks and impacts to the examination results.

Regarding existing forensic tools, the viaForensics company developed a free tool to law

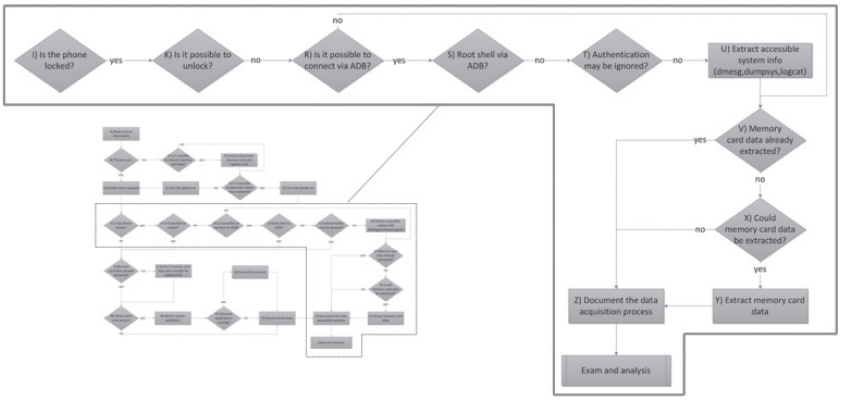

Figure 9. Processes of data acquisition of an Android smartphone with access control.

enforcement agencies called “Android Forensic Logical Application” (AFLogical) [20], whose goal is to extract information from Android smartphones. In addition, recently the commercial tool viaExtract was released and, according to viaForensics, it has more consistent and relevant features, such as generating reports. Another very useful tool is the "Cellebrite UFED", whose version 1.1.7.5, released in July of 2011, carries out physical extraction from a few models without the need of "super user" privileges. The same tool has a plugin to view Android's SQLite databases and other application files, such as Gmail, SMS, MMS and contacts.

C. Smartphone with access control

In the likely event the Android smartphone has access control, such as a password or tactile pattern, there are still techniques to be used to access the device.

According to NIST [17], there are three ways of gaining access to locked devices. The first one is the investigative method, whereby the researcher seeks possible passwords in the place where the smartphone was seized or even interview its alleged owner so that he cooperates voluntarily by providing his password. Another way is to gain access via hardware, when the analyst performs a research on that specific given model to determine whether it is possible to perform a non-destructive procedure in order to access device data. In this sense, one may request support from manufacturers and authorized service centers. Finally, there are software access methods that, even though they depend on the handset model and Android version, are usually the easiest ways and can be applied in the forensic analyst's own test environment. Figure 9 illustrates the process of extracting data from an Android device with access control enabled.

To access the system, the analyst must do it the least intrusive manner possible in order to avoid compromising the evidence. If the password or the tactile pattern has been obtained when the device was seized, those should be readily tested. Alternatively, one may use the technique to find the tactile pattern by means of examining the smudge left on the device screen [21], before attempting any other way to bypass access control, preventing screen contamination.

If the analyst does not succeed, he should check if the Android is configured to accept USB debugging connections using a tool available in the SDK, the ADB. If he succeeds, he attempts to obtain "super user" access credentials to resume the acquisition process, the same way that it would be performed in cases which the mobile device was not locked, because with such permissions, one could get all the stored data in the device, as described previously.

Even when there is no "super user" access to the handset, it is still possible to install applications through the ADB tool to overcome the access control system. The technique described by Thomas Cannon [22] is to install the "Screen

Lock Bypass" application, available in the Android Market. In this technique, one needs the Google account's password to be saved in the Android device, as well as Internet access to be enabled, which is considered inadvisable. In this sense, it is recommended that the application is downloaded from another Android device and then installed via ADB on the examined mobile device. Thus, it is possible to perform the screen unlock using Cannon's technique without the need of having the device's Google account password or connecting it to the web. Figure 10 shows Cannon's application installation, as well as its activation, which depends on the installation of any other application, to perform access control unlocking.

```
C:\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
Permission denied
$ exit
...

C:\android-sdk\platform-tools>adb -s 040140611301E014 install
screenlockbypass.apk
224 KB/s (22797 bytes in 0.100s)
        pkg: /data/local/tmp/screenlockbypass.apk
Success

C:\android-sdk\platform-tools>adb -s 040140611301E014 install
AndroidForensics.apk
716 KB/s (31558 bytes in 0.046s)
        pkg: /data/local/tmp/AndroidForensics.apk
Success
```

Figure 10. Connection via ADB, root access check and application installation in order to ignore access control.

In situations where it is not possible to bypass the authentication system or USB debugging access is disabled, it is left to the analyst to copy the data contained in the removable memory card that may be installed on the handset. In those situations, it is very important to report the impossibility to access the device with the used procedures. In addition, if there is another technique that may be applied, be it more invasive or complex, that fact should be informed to whoever requested the exams. Consequently, the implications of applying such techniques should be discussed, considering the risks to the given situation, such as permanent damages to the examined smartphone.

D. Acquisition documentation

It is recommended that all the techniques and procedures used by the analyst should be documented, in order to facilitate the examination and analysis of extracted data. Regardless of the path followed by the expert in the workflow illustrated in Figure 3, the process should be recorded, enabling auditability and reliability of the procedures performed by the expert analyst.

The analyst should be careful to register the hash codes of the data generated and extracted during the acquisition process, as well as state in his report any caveats that he considers important to carry out the examination and analysis stage, like an e-mail or SMS received before the smartphone has been isolated from telecommunication networks or even the existence of applications that contain information stored in servers on the Internet, such as cloud computing.

The forensic expert, while executing his activities, should consider that the better reported the acquisition process, the more trust will be given to the examination results. The simple condition of the processes be well documented is the first step to conduct an impartial, clear and objective data analysis.

V. Examination and Analysis Process

The steps to examine and analyze the data of a smartphone with the Android system are illustrated in the workflow in Figure 11.

A. Goals definition

Before beginning the analysis of data extracted in the previous step, the forensic analyst should be concerned about the study objectives, based on what is being investigated. This definition is important because, depending on what is being investigated, the examination on the extracted data may follow different paradigms. For example, the focus may be just pictures and videos, contacts or geolocation.

B. Smartphone individualization

After the examination goals have been defined, the expert should seek information that can point to the device's owner in extracted

data and even in the very smartphone when necessary, individualizing it. Searches are performed on the extracted data, such as the Google account username, e-mails, IM users, notes, calendar, digital business cards, among others. The phone individualization determines who the user of the device is, so one can link the evidence found through the analysis to a suspect in an unquestionable way.

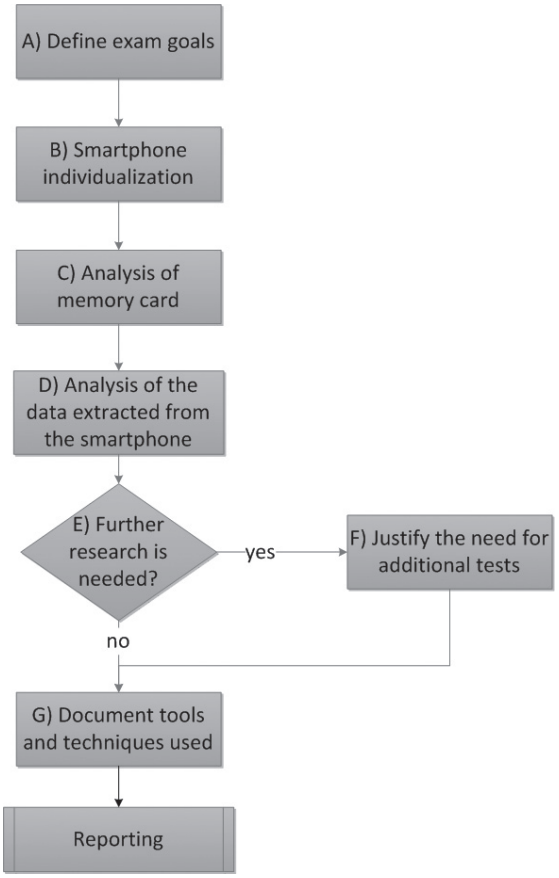

Figure 11. Workflow of the examination and analysis process.

C. Device data analysis

The analysis starts with the data extracted from memory cards. With a memory card image, which was obtained in the acquisition phase, it is possible to use forensic tools commonly used for computers for viewing the file structure, search by keywords, search for regular expressions, viewing photos and videos or examining to try to achieve a specified objective.

Thereafter, smartphone data examination may vary depending on how it was obtained. If they have been extracted using a forensic tool, one must analyze the produced output, observing the report files generated and retrieved. Specific tools for a particular platform usually can achieve good results, since they simulate a manual extraction, automating the process. However, in the acquisition phase, the analyst must have performed a comparison of what was extracted by the application with the information contained on the phone, complementing the forensic report generated by the tool.

If the extracted data were obtained from a system image using a super user access, the examiner could make use of hex editors and forensic tools to analyze the memory card and other forensic techniques, in order to perform the analysis. One could also rely on ".dex" file disassemblers to audit installed applications.

In order to analyze the databases extracted from the phone's memory card or internal memory, the analyst must use the SQLite software, since the Android platform adopted this relational database as default. The analysis for the SQLite files is very important, since almost all data stored by applications are in that database management system. Thus, depending on the situation, it is possible, for instance, to get information about the maps cached by Google Maps Navigation [23].

VI. The proposed method validation

The proposed method was tested by using a sample containing six smartphones that had the Android OS. Among those handsets, four different scenarios were identified, summed up and presented in Table 1.

Will be considered, as the examination objective, the extraction of information deemed as relevant by the expert. This could be text messages, call logs, emails, images, videos or anything else that could be fed to the investigative procedure.

TABLE II. Scenarios used to validate the proposed method.

Scenario	Turned on	Removable card	Locked	Unlockable	Super user
1st Scenario (Motorola Milestone II A953)	No	Yes	Yes	Yes	No
2nd Scenario (Sony Ericson Xperia X10 miniPro)	Yes	No	No	Does not apply	No
3rd Scenario (Motorola Defy) (Samsung Galaxy S 9000[a])	No	Yes	No	Does not apply	Yes
4th Scenario (Motorola I1) (Motorola Milestone A853)	No	Yes	No	Does not apply	No

[a] In addition to the removable microSD card, that phone has a built-in memory card which is not removable

A. 1st scenario

In the first scenario, as the device was turned off, first its memory card was removed and mirrored. Then, the memory card holding the copy was inserted into the handset. Subsequently, the smartphone was switched on and set immediately in flight mode. It was noticed that the cell phone was locked, but its USB debugging access was enabled. By using the ADB tool, a shell was obtained but that there were no "super user" permissions available, preventing mirroring system partitions. However, from the ADB, it was possible to install the "Screen Lock Bypass" application [22], which was used to unlock the device, as well as the "Logical Android Application Forensics" [20], a data extraction tool. In addition, the extracted data were visually inspected.

In the exam and analysis stage, the cell phone was individualized through its Google account, since the investigated person used his own name in his email account. It was also possible to obtain images from the memory card data. Some of them appeared to be family photos. It was not able to retrieve metadata from the photos though, such as GPS coordinates of where they were taken.

From the data extracted from the smartphone, it was possible to get the phonebook contacts. In addition, it was observed that the user made little use of SMS messages and often used the calendar for his appointment records, like when he supposedly was at the gym, theater and pharmacy. Besides, 500 records of received and missed calls were obtained. Since this was an average user's smartphone, without deep platform knowledge, there was no information available that would justify further investigation. Finally, used tools and techniques were documented.

B. 2nd scenario

In the following scenario, the smartphone was not locked and was put into flight mode in order to isolate it from the network. The device had a memory card that was not removable. The card data were mirrored (copied entirely), and its own memory was used to extract its information by the "Logical Android Application Forensics" forensic software. Afterwards, data were extracted the same way as the previous scenario.

In this scenario, it was observed that there were, in the Bluetooth memory card folder, 60 "vcf" files, known as business cards. They had probably been sent to the phone via Bluetooth and then were incorporated to the phone's contacts. There was a file named "Home.vcf" stored with a landline number, which possibly indicated the residence of the smartphone owner. Moving on to a deeper data analysis from the memory card, two photographs were found, which had geographic coordinates metadata.

In addition, several received and missed call records were obtained, as well as contacts and text information. As in the first scenario, since this was a smartphone from an average user, there was no information available that would justify further investigation. Eventually, the tools and techniques that were used were documented.

C. 3rd scenario

The same way as done in the first scenario, in the third, the memory card was removed and replaced by a mirror, since the device was received turned off. Later, the smartphone was turned on and immediately put into flight mode. It was noticed that the mobile device was unlocked and also had a second memory card embedded. That memory card was also mirrored. The smartphone had the "Superuser" application, which provides super user credentials. Then, the USB debug mode was enabled, an ADB was established, obtaining a shell with "super user" permissions to carry out the smartphone partitions mirroring (system, userdata and cache). A logical extraction of important files was also performed, as the ones related to applications, including databases and system configuration files [24]. RAM data were not copied, because the handset was received switched off and the analysts considered unnecessary to perform such procedure. Then, the Cellebrite UFED System 1.1.7 tool was used to extract forensic data from the phone, followed by visual inspection to complement the extracted data.

About the exam and analysis stage, the Motorola Defy results will be presented. The system, cache and userdata partitions mirrors (complete copies), were examined in FTK [25] with the data carving option, since there was no support for YAFFS2, which limited the analysis.

From the logical analysis of the data copied in the directory /data/system, it was possible to obtain the list of applications installed on the system (file "package.list") and the account set up for the Google phone with encrypted password ("accounts.db" file"). In the /data/misc folder, Wi-Fi settings and WPA2 passphrases were found stored in clear text in the "wpa_supplicant.conf" file.

Examining the cache files retrieved from the directory /data/data, payment and money transfer receipts, current account statements and credit card limits were found in the "br.com.bb.android" application data. It was noted that the phone had the "Seek Droid" ("org.gtmedia.seekdroid") application, which allows location, blocking and data deletion remotely through the www.seekdroid.com web site. In this application's installation directory, the "prefs.xml" file was found, which contained information about its configuration, username and password. The "Gtalk" application provided, in the "talk.db" file, chat history and friends list. Information about sent and received e-mails, along with date, times, sender and recipient were obtained from the "mailstore.<googleusername>@gmail.com.db" file of the "com.google.android.gm" application. SMS messages were stored in the "mmssms.db" file of the "com.android.providers.telephony" application. Calendar events were found in the "calendar.db" file of the "com.android.providers.calendar" application. From the "webview.db" file of the "com.android.browser" application, it was found that the phone user had logged on websites such as Facebook (http://m.facebook.com), Yahoo (http://m.login.yahoo) and MercadoLivre (https://www.mercadolivre.com). From the "DropboxAccountPrefs.xml" file of the "com.dropbox.android" application, it was possible to obtain the configured user name, as well as the "db.db" file which had a directories and files list, with their respective sizes. The system configurations were found in the "settings.db" file of the "com.android.provider.settings" application. Much more information can be obtained from the cache and database files, when the expert must further examine the device to achieve his goal.

D. 4th scenario

Last but not least, in the fourth scenario, the memory card was removed, mirrored and replaced while the device was still turned off. Then, the phone was turned on and immediately put into flight mode. The phone was unlocked. Thus, the Cellebrite UFED System 1.1.7 tool was used to extract forensic data from the phone, with subsequent visual inspection to complement the extracted data. Then, the data were examined and analyzed and the procedures were documented.

The procedures cited in the method could be directly translated into actions performed onto the examined devices. Thus, it was possible to perform data acquisition of every tested smartphone, demonstrating the suitability and validity of the proposed method for each encountered scenario.

VII. Conclusion

The Android smartphone platform is already the most present among mobile communication devices. However, the existing approaches to forensic examine cell phones and computers are not completely adequate to the peculiarities of that class of devices. Moreover, the existing models of forensic analysis on cell phones do not consider the peculiarities of each platform.

A specific method was proposed to address data acquisition of devices that use the Android Platform, taking into account operating system characteristics, its most popular applications and hardware features.

By means of defining an Android system data acquisition method, it was possible to foresee the difficulties forensic experts might face, preparing them to perform an entire evidence acquisition, given the situation the handset was forwarded, avoiding mishaps in the data extraction process and missing forensic evidence.

The method was proposed in a broad fashion, so that the techniques, procedures and specific tools chosen by the analyst during the workflow do not interfere with its application. So, as new techniques arise, with different approaches to perform a given task, such as unlocking the device, bypassing access control or mirroring partitions, they will be covered by the proposed method, which focuses on the result that each activity produces.

The proposed method was validated by its application onto the examination of six Android smartphones, which were grouped into four scenarios, involving different situations that an analyst might encounter.

For future work, it is suggested that the method be validated for the Android 3, evaluating its effectiveness in the Google system for tablet devices, as well as in Android 4, making the adjustments that may be required. Another interesting work to be developed would be the creation of a forensic tool that supported the YAFFS2 file system, focused on NAND flash memory, facilitating data extraction and access and also mounting images from those storage media.

Acknowledgements

This work was developed with institutional support by the Brazilian Federal Police (DPF) and with financial aid from the National Public Security and Citizenship Program (PRONASCI), an initiative led by the Ministry of Justice. The studies were carried out under the supervision of Professors from the Electrical Engineering Department at University of Brasilia, who contributed to directing the efforts and producing high level scientific knowledge

References

[1] CANALYS. Android takes almost 50% share of worldwide smartphone market. **Canalys web site**, 2011. Avaiable at: <http://www.canalys.com/newsroom/android-takes-almost-50-share-worldwide-smartphone-market>. Accessed in: 03 August 2011.

[2] PETTEY, C.; STEVENS, H. Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. **Gartner web site**, 2011. Avaiable at: <http://www.gartner.com/it/page.jsp?id=1848514>. Accessed in: 16 November 2011.

[3] ROSSI, M. Internal Forensic Acquisition for Mobile Equipments, n. IEEE, 2008.

[4] ASSOCIATION OF CHIEF POLICE OFFICERS. **Good Practice Guide for Computer-Based Electronic Evidence - Version 4.0**. [S.l.]. 2008.

[5] FARLEY, T. Mobile Telephone History. **Tlektronikk**, v. 3, p. 22 to 34, April 2005.

[6] CASSAVO, L. In Pictures: A History of Cell Phones. **PCWorld**, 7 maio 2007. Avaiable at: <http://www.pcworld.com/article/131450/in_pictures_a_history_of_cell_phones.html>. Accessed in: 22 March 2011.

[7] SPECKMANN, B. **The Android mobile platform**. [S.l.]: Eastern Michigan University, Department of Computer Science, 2008.

[8] CAVALEIRO, D. Nokia e Microsoft confirmam parceria para enfrentar Apple e Google. **Jornal de Negócios**, 11 fevereiro 2011. Avaiable at: <http://www.jornaldenegocios.pt/home.php?template=SHOWNEWS_V2&id=468017>. Accessed in: 22 March 2011.

[9] GADHAVI, B. **Analysis of the Emerging Android Market**. The Faculty of the Department of General Engineering, San Jose State University. [S.l.], p. 88. 2010.

[10] GOOGLE INC. What is Android? **Android Developers**, 2011. Avaiable at: <http://developer.android.com/guide/basics/what-is-android.html>. Accessed in: 8 April 2011.

[11] HASHIMI, S.; KOMATINENI, S.; MACLEAN, D. **Pro Android 2**. 1st Editon. ed. [S.l.]: Apress, 2010. ISBN 978-1-4302-2659-8.

[12] EHRINGER, D. The Dalvik Virtual Machine Architecture. **David Ehringer**, março 2008. Avaiable at: <http://davidehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf>. Accessed in: 17 February 2011.

[13] BURNETTE, E. **Hello, Android**. [S.l.]: Pragmatic Bookshelf, 2008. ISBN 978-1-934356-17-3.

[14] HOOG, A. **Android Forensics - Investigation, Analisys and Mobile Security for Google Android**. 1st Edition. ed. [S.l.]: Syngress, 2011.

[15] GOOGLE INC. Android Fundamentals. **Android Developers**, 2011. Avaiable at: <http://developer.android.com/guide/topics/fundamentals.html>. Accessed in: 17 March 2011.

[16] SQLITE. About SQLite. **SQLite**, 2011. Avaiable at: <http://www.sqlite.org/about.html>. Accessed in: 5 April 2011.

[17] JANSEN, W.; AYERS, R. **Guidelines on Cell Phone Forensics - Recomendations of the National Institute of Standards and Technology**. [S.l.]. 2007.

[18] CANNON, T. Android Reverse Engineering. **Thomas Cannon**, 2010. Avaiable at: <http://thomascannon.net/projects/android-reversing/>. Accessed in: 23 March 2011.

[19] HOOG, A. **Android Forensics - Investigation, Analisys and Mobile Security for Google Android**. 1st. ed. [S.l.]: Syngress, 2011.

[20] VIAFORENSICS. Android Forensics Logical Application (LE Restricted). **Sítio da viaForensics**, 2011. Avaiable at: <http://viaforensics.com/android-forensics/android-forensics-logical-application-le-restricted.html>. Accessed in: 03 August 2011.

[21] AVIV, A. J. et al. **Smudge Attacks on Smartphone Touch Screens**. 4th Workshop on Offensive Technologies. Washington, DC: [s.n.]. 2010.

[22] CANNON, T. Android Lock Screen Bypass. **Thomas Cannon**, 2011. Avaiable at: <http://thomascannon.net/blog/2011/02/android-lock-screen-bypass/>. Accessed in: 23 March 2011.

[23] HOOG, A. Google Maps Navigation - com.google.apps.maps. **Via Forensics**, 2010. Avaiable at: <http://viaforensics.com/wiki/doku.php?id=aflogical:com.google.android.apps.maps>. Accessed in: 20 April 2011.

[24] LESSARD, J.; KESSLER, G. C. Android Forensics: Simplifying Cell Phone Examinations. **Small Scale Digital Device Forensics Journal**, September 2010.

[25] ACCESSDATA. Forensic Toolkit (FTK) Computer Forensics Software. **Sítio da internet da AccessData**, 2011. Avaiable at: <http://accessdata.com/products/computer-forensics/ftk>. Accessed in: 10 October 2011.

André Morum de Lima Simão has a bachelor degree in Computer Science from Catholic University of Brasília (2000), a postgraduate degree in Information Security Management from University of Brasilia (2002) and obteined his masters degree in Computer Forensics and Information Security in the Electrical Engineering Department at University of Brasilia (2011). He joined the team of forensic experts of the Brazilian Federal Police of Brazil in 2005, where he has been conducting activities in the area of computer forensics.

Fabio Caus Sicoli has a bachelor degree in Computer Science from University of Brasilia (2004) and a postgraduate degree in Cryptography and Network Security from Fluminense Federal University (2010). He is a masters student in Computer Forensics and Information Security in the Electrical Engineering Department at University of Brasilia. He has been working as a forensic expert in computer crimes in the Brazilian Federal Police for the last six years.

Laerte Peotta de Melo is graduated in Electrical Eng. with emphasis in Electronics by Mackenzie University (1996), specialization in security of computer networks by Católica University (2004), computer forensics expert by Universidade Federal do Ceará (2007), Master's degree in Electrical Engineer by Brasília University (2008). He is currently on pursuit PhD by Brasília University (2008).

Flavio Elias Gomes de Deus received his BS in Electrical Engineering from Universidade Federal do Goiás, in 1998, MS in Electrical Engineering from Universidade de Brasília, in 2001, and Ph.D. in Electrical Engineering from the Universidade de Brasília, in 2006. He was also Visiting Scholar in Information Science and Telecommunications at University of Pittsburgh, USA, from 2004 to 2005. He is currently Associate Professor in the Department of Electrical Engineering, Universidade de Brasília, Brazil. His research interests include information technologies, information and network security, fault tolerant systems, software development process, among other related topics.

R. T. de Sousa, Jr., was born in Campina Grande – PB, Brazil, on June 24, 1961. He received his B.S. degree in Electrical Engineering, from the Federal University of Paraíba – UFPB, Campina Grande – PB, Brazil, in 1984, and got his Doctorate Degree in Telecommunications, from the University of Rennes 1, Rennes, France, in 1988. His professional experience includes technological consulting for private organizations and the Brazilian Federal Government. His sabatical year 2006-2007 was with the Networks and Information Systems Security Group, at Ecole Superiéure d´Electricité, Rennes, France. He is currently an Associate Professor with the Department of Electrical Engineering, at the University of Brasília, Brazil, and his current research interest is trust and security in information systems and networks.